**Введение**

На курсовое проектирование была поставлена задача, разработать учебное ПО, визуализирующее доступ к информации, размещённой в кеш-памяти.

Цель курсового проекта заключается в том, чтобы помочь учащимся сильнее углубиться в данную тему и, при помощи визуализации, легче освоить материал.

Создаваемая программа будет рассчитана на любого рода пользователей. Применить данную программу смогут не только учащиеся, но и просто люди, заинтересовавшиеся в данной теме.

Далее приведём краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение её организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создаётся данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет чётко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут чётко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

**1. Анализ задачи**

**1.1 Постановка задачи**

Темой данного курсового проекта является «Разработка учебного ПО, визуализирующего доступ к информации, размещённой в кэш-памяти»

Кэш-память является важной и неотъемлемой частью современных компьютерных систем. В нашем быстро развивающемся мире, где время становится все более ценным ресурсом, кэш-память играет решающую роль в оптимизации работы компьютера и повышении его производительности. Она служит промежуточным буфером между процессором и оперативной памятью, ускоряя доступ к часто используемым данным.

Кэш-память сокращает время доступа к данным, уменьшая задержки при обращении к оперативной памяти. Расположенная ближе к процессору, она быстро поставляет данные, которые часто запрашиваются, повышая общую производительность системы. Кроме того, кэш-память улучшает эффективность использования оперативной памяти, позволяя процессору быстро получать доступ к сохраненным данным и уменьшая операции чтения и записи в оперативную память. Она также оптимизирует работу многопоточных и многопроцессорных систем, улучшая распределение данных и повышая производительность системы в целом.

Аналогов для такого приложения можно привести не так уж и много. В основном в сети интернет находятся различного рода статьи по данной тематике, сочинения других людей. В более редких случаях – ознакомительный материал с темами для изучения и повторения знаний.

Периодичность использования данного программного продукта неограниченна. Пользователь может в любое время установить, а также удалить приложение после полного изучения материала по данным темам и прохождения теста.

Цель данного курсового проекта – разработать программный продукт, который будет помогать лучше разобраться в теме, закрепить полученные знания и подготовиться к занятию, а так же он поможет преподавателю более доступно донести информацию до учащихся.

Основу программы возьмём из соответствующих источников (сайтов, статей), а наиболее полную, достоверную и точную информацию получим от преподавателя. При реализации проекта необходимо его сделать простым, максимально доступным и понятным для пользователей.

Приложение должно располагать своим интерфейсом и функционалом к пользователю, чтобы ему было комфортно.

Данный проект должен стать общедоступным для всех пользователей. В поставленной задаче необходимо реализовать максимально простой пользовательский интерфейс, позволяющий использовать проект пользователю, не обладающему дополнительными знаниями электронно-вычислительной техники.

Разрабатываемый программный продукт должен позволять выполнять следующие действия:

* обеспечить быстрый поиск необходимой информации;
* прохождение теста для закрепления знаний;
* просмотр справки;
* просмотр практических заданий;

В разрабатываемой программе будут использоваться три вида данных: входная, выходная и постоянная информация.

К входной информации можно отнести вводимые пользователем значения,

например, ответы на вопросы при прохождения теста. К выходной – результат прохождения теста.

Постоянной информацией в проекте будут являться формы с визуализацией, картинки, теоретический материал и др.

Программный продукт предоставляет функционал для следующего ряда пользователей: администратор – организовывает работу программы, гость, который хочет изучить данную в проекте тему.

**1.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда разработки Delphi 11, так как это самое удобная и доступная среда разработки на данный момент. Delphi 11 – язык программирования, относящийся к классу RAD(Rapid Application Development – «Средство быстрой разработки приложений») средств CASE –тeхнологии. Delphi 11 сделал разработку приложений для Windows быстрым и приятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi 11.

Интерфейс Windows обеспечивает полное перенесение CASE-технологии в интегральную систему поддержки работ по созданию прикладной системы на всех фазах жизненного цикла работы и проектирования системы.

Delphi 11 обладает широким набором возможностей, начиная от проектировщика форм и заканчивая поддержкой всех форматов популярных баз данных. Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, программы и даже диалоговые панели. Работая в Windows, можно видеть одинаковые «объекты» во многих разнообразных приложениях. Диалоговые панели (например, Choose File и Save File) являются примерами многократно-используемых компонентов, встроенных непосредственно в Delphi 11, который позволяет приспособить эти компоненты к имеющийся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно-определенные визуальные и не визуальные объекты.

Три основные части разработки интерфейса следующие: проектирование панели, проектирование диалога и представление окон. Для общего пользовательского доступа также должны учитываться условия применения архитектуры прикладных систем.

Сегодня появилась реальная возможность с помощью моделирования на современных многофункциональных средствах обработки и отображения информации таких как Delphi 11 конкретизировать тип и характеристики используемых информационных моделей, выявить основные особенности будущей деятельности операторов, сформулировать требования к параметрам аппаратно-программных средств интерфейса взаимодействия и т.д. Delphi 11 позволяет создать различные виды программ: консольные приложения, оконные приложения, приложения для работы с Интернетом и базами данных. То есть, Delphi 11 является не только средствами для работы с языком программирования Паскаль, но дополнительные инструменты, призванные для максимального упрощения и ускорения создание приложений.

К дополнительным инструментам можно отнести визуальный редактор форм, благодаря которому можно с легкостью создать полноценную программу, и другие визуальные составляющие разработки программного обеспечения. С Delphi вам не нужно вручную просчитывать расположение каждого элемента интерфейса пользователя, поэтому при разработке программы значительно экономится время.

Выгоды от проектирования в среде Windows с помощью Delphi 11:

* устраняется необходимость в повторном вводе данных;
* обеспечивается согласованность проекта и его реализации;
* увеличивается производительность разработки и переносимость программ.

Ни одно серьезное программное обеспечение не обходится без модуля справочной информации и руководства пользователя. Это придает программе законченный вид и показывает заботу о пользователе. HelpAndManual8 – легкий в использовании и функциональный инструмент, упрощающий создание справочных файлов Windows, печать справочных руководств и документации в целом. Программа имеет интуитивно понятный интерфейс. Все созданные проекты можно сохранить в различных форматах:HTML Help, Winhelp и MS Help 2.0 / Visual Studio Help, Browser-based Help, PDF и Word RTF, а также печатной документации при использовании одного и того же проекта. В основном окне программы содержатся оглавление (в виде древовидного списка) и текстовый редактор. Это дает возможность легко ориентироваться в оглавлении, редактировать или перемещать разделы справки без каких-либо проблем. Также утилита позволяет конвертировать help-файлы из одного формата в другой. Кроме приложений для работы с текстом в данном продукте содержатся утилиты для создания скриншотов и редактирования графических файлов.

Для создания инсталлятора будет использоваться мощное и удобное средство - Smart Install Maker. Программа обладает удобным и интуитивно понятным интерфейсом, а также полным набором необходимых функций для создания профессиональных инсталляторов с минимальным размером, высокой степенью сжатия файлов и приятным интерфейсом.

Помимо стандартного минимума, Smart Install Maker позволяет редактировать системный реестр и INI-файлы, создавать программные ярлыки, запускать ассоциируемые и исполняемые файлы, регистрировать новые шрифты и ActiveX компоненты, отображать тексты информации и лицензионного соглашения. Также, с помощью этой утилиты, можно создать мультиязыковые инсталляторы с поддержкой более 20-ти популярных языков мира.

Easy Paint tool SAI - это популярная среди художников программа для рисования (в частности для проработки интерфейса и игры «одевалка»). Удобна своей простотой и в то же время большим функционалом. Программа очень сильно распространена в использовании во всем мире и предпочитается рисующими людьми как один из самых удобных инструментов для создания рисунков. В нем можно сохранить файлы картинок во множестве вариантов (.png, .jpg, .bmp и так далее).

Microsoft Word 2019 – редактор текста для написания документации.

Минимальные системные требования:

- 1.8 GHz процессор;

- 2 GB RAM;

- от 500 MB доступного места на жёстком диске.

Рекомендуемые системные требования:

- двухъядерный AMD Athlon или аналогичный процессор Intel (или более

производительный);

- 4 GB RAM;

- от 500 MB доступного места на жёстком диске.

Как видно разрабатываемое приложение не очень требовательно к аппаратным ресурсам, что, является большим плюсом.

**1.3 Требования к приложению**

На этапе исследования предметной области был установлен целый ряд требований, которые предъявляются к разрабатываемой программе.

По этой причине особенно актуальной становится разработка программного электронного продукта, который способствовал бы и давал возможность вспомнить и повторить информацию из теоретического курса. Чтобы эта информация лучше усваивалась необходимо подобрать правильный интерфейс.

Требования к интерфейсу: в связи с частым использованием программы она должна быть с приятной цветовой гаммой и понятной для пользователя. Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информация, отображённой на экране. Объекты (рисунки и символы) не должны быть слишком мелкие. Все окна приложения по возможности должны помещаться на экран полностью, так как использование в процессе работы полос прокруток достаточно неудобно.

На одной форме нельзя допускать избытка и нагромождения данных. Формы должны быть эффектно оформлены согласно тематике разрабатываемого проекта.

Требования к надежности: специальных требований к надежности не предъявляется. Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбое операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на конечный результат.

**2. Проектирование задачи**

**2.1 Организация данных**

Проектирование задачи – это очень важный и ответственный этап в разработке любого приложения.

Важным является он вследствие того, что методы, по средствам которых пользователь управляет формами, построены на высокой степени специализации каждого из компонентов.

Необходимым условием при разработке данного приложения является описание организации данных, т.е. логическая и физическая структура данных

в контексте среды разработки. В разрабатываемой программе будут использоваться три вида данных.

Первым видом являются данные, которые будут введены разработчиком на этапе реализации задачи. Сюда можно отнести изображения (иконки), описание, исходные коды и др.

Вторым видом данных, используемых в программе, является вводимая пользователем информация. Входной информацией в разрабатываемой программе будут являться данные. Подходящей структурой данных для хранения элементов является: динамический одномерный массив.

Третьим видом данных является результат программы –отсортированные пользовательские данные. Его также относят к отдельному виду, так как ни пользователь, ни разработчик его не вводят, а программа сама получает его в результате выполнения определенных действий.

Таким образом, организация данных является важной задачей при разработке данной и любой программы.

**2.2 Процессы**

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно

должна заниматься разрабатываемая приложение. Главной задачей будет являться: усвоение материала по теме, его повторение и подготовка к учебным занятиям, а так же это интерактивный способ подачи информации для преподавателя .

Для реализации задач будут использоваться процедуры. С помощью процедуры будет осуществляться переход на текстовый файл в формате .txt.

Теория, визуализация и тест будут генерироваться на основе выбора пункта в объекте TPageControl. Далее после того, как будет прочитан теоретический материал и выполнена практическая часть, можно будет пройти тест для проверки своих знаний и получить оценку своих знаний.

**2.3 Описание внешнего пользовательского интерфейса**

Важным при выполнении курсового проекта является организация диалога между, пользователем и самой программой. Во многом это зависит от того, как программист разработает данную программу, какие компоненты будут использованы и какие методы будут автоматизированы. Во-первых, особое внимание следует уделить интерфейсу. Разработчик должен так организовать внешний вид своей программы, чтобы пользователь понял, что от него требуется.

Для организации эффективной работы пользователя нужно создать целостное приложение данной предметной области, в которой все компоненты приложения будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Приложение должно позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами.

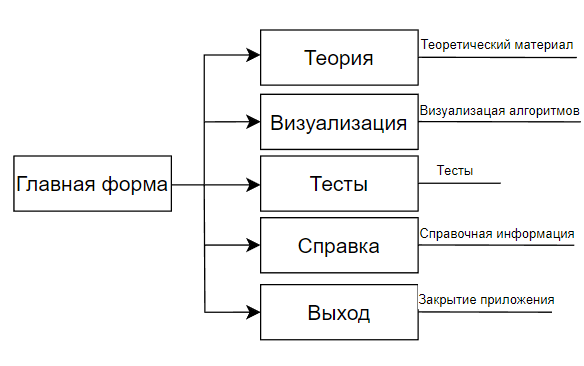
Ниже на рисунке 1 представлена система меню и организация навигации между окнами программы:

Рисунок 1- схема навигации между окнами программы

**3. Реализация**

**3.1 Реализация проекта**

Данный курсовой проект содержит 6 модулей. Далее рассмотрим назначение каждого модуля:

Unit 1 – содержит визуализацию полностью ассоциативной кэш-памяти

Unit 2 – содержит визуализацию кэш-памяти с прямым доступом

Unit 3 – содержит визуализацию Двухканально наборно-ассоциативной кэш-памяти

Unit 4- является главным меню для навигации по проекту

Unit 5- содержит теоретический материал о кэш-памяти

Unit 6- Объединяет визуализацию всех 3-ех видов кэша

Unit 7-тестовые задания.

Unit 8- является заставкой в курсовом проекте;

**3.1.1 Структура и описание процедур и функций пользователя**

Описание разработанных процедур находятся в таблице 1.

Таблица 1 – Процедуры и функции

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя процедуры (функции) | | В каком модуле находится | | За каким компонентом закреплена | | | Назначение | |
| 1 | | 2 | | 3 | | | 4 | |
| procedure TCache\_1.Button1Click (Sender: TObject); | | Unit 1 | | Button1 | | | Запуск визуализации | |
| procedure TCache\_1.FormCreate (Sender: TObject); | | Unit 1 | | TCache\_1 | | | Создание и заполнение объектов формы | |
| procedure TCache\_1.StringGrid1DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 1 | | Stringgrid1 | | | Изменение содержимого ячеек Stringgrid | |
| procedure TCache\_2.FormCreate (Sender: TObject); | | Unit2 | | TCache\_2 | | Создание и заполнение объектов на форме | | |
| Продолжение таблицы 1 | | | | | | | | |
| 1 | | 2 | | 3 | | 4 | | |
| procedure TCache\_2.StringGrid1DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 2 | | Stringgrid1 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_2.StringGrid2DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 2 | | Stringgrid2 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_2.Stringgrid3DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 2 | | Stringgrid3 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_3.Button1Click (Sender: TObject); | | Unit 3 | | Button1 | | Запуск визуализации | | |
| procedure TCache\_3.FormCreate (Sender: TObject); | | Unit 3 | | TCache\_3 | | Создание и заполнение объектов | | |
| procedure TCache\_3.StringGrid1DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid1 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_3.StringGrid2DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid2 | | Изменение содержимого ячеек Stringgrid | | |
| Procedure TCache\_3.StringGrid3DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid3 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_3.StringGrid4DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid4 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_3.StringGrid5DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid5 | | Изменение содержимого ячеек Stringgrid | | |
| procedure TCache\_3.StrigGrid6DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid6 | | Изменение содержимого ячеек Stringgrid | | |
| Procedure TCache\_3.StrinGrid7DrawCell (Sender: TObject; ACol, ARow: Integer;  Rect: TRect; State: TGridDrawState); | | Unit 3 | | Stringgrid7 | | Изменение содержимого ячеек Stringgrid | | |
| Продолжение таблицы 1 | | | | | | | | |
| 1 | | 2 | | 3 | | 4 | | |
| procedure TMyMenu.Button1Click (Sender: TObject); | | Unit 4 | | Button1 | | Переход на форму с теорией | | |
| procedure TMyMenu.Button2Click (Sender: TObject); | | Unit 4 | | Button2 | | Переход на форму с визуализацией | | |
| procedure TMyMenu.Button3Click (Sender: TObject); | | Unit 4 | | Button3 | | Переход на форму с тестом | | |
| procedure TMyMenu.Button4Click (Sender: TObject); | | Unit 4 | | Button4 | | Выход из программы | | |
| procedure TMyMenu.N1Click(Sender: TObject); | | Unit 4 | | Main | | Выход из программы | | |
| procedure TMyMenu.N2Click(Sender: TObject); | | Unit 4 | | Main | | Открытие справки | | |
| procedure TTeory.FormCreate(Sender: TObject); | | Unit 5 | | TTeory | | Загрузка информации из webbrowser | | |
| procedure TTeory.N1Click(Sender: TObject); | | Unit 5 | | MainMenu1 | | Выход из программы | | |
| procedure TTeory.N5Click(Sender: TObject); | | Unit 5 | | MainMenu1 | | Возврат на главную форму | | |
| procedure TTeory.N2Click(Sender: TObject); | | Unit 5 | | MainMenu1 | | Открытие справки | | |
| procedure TVizualization.FormCreate (Sender: TObject); | | Unit 6 | | TVizualization | | Подключение других форм в pagecontrol | | |
| procedure TVizualization.N1Click(Sender: TObject); | | Unit 6 | | MainMenu1 | | Закрытие программы | | |
| procedure TVizualization.N2Click(Sender: TObject); | | Unit 6 | | MainMenu1 | | Открытие справки | | |
| procedure TVizualization.N3Click(Sender: TObject); | | Unit 6 | | MainMenu1 | | Переход на главную форму | | |
|  | |  | |  | |  | | |

**3.1.2 Описание использованных компонентов**

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2- Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме  расположен | Назначение |
| TMainMenu | Form2, Form4, Form6 | Используется для создания главного меню проекта |
| TImage | Form2, Form3, Form4, Form5, Form6, Form7 | Используется как фон |
| TTimer | Form7 | Используется для определения длительности теста |
| TLabel | Form2, Form3, Form4, Form5, Form6, Form7 | Отображение надписей на форме |
| TButton | Form2, Form3, Form4, Form5, Form6, Form7 | Используется для взаимодействия  пользователя с программой: для начала теста, закрытие программы, перехода на другую формы и т.д. |
| TCheckListBox | Form2 | Используется для хранения теоретического материала |
| TRadioGroup | Form7 | Используется для выбора варианта ответа в тесте |
| TProgressbar | Form7 | Используется для полосы времени |
| TEdit | Form5 | Используется для ввода информации пользователя |
| Ttimes | Form7 | Используется, как таймер |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |  |
| --- | --- | --- |
| Имя файла | Назначение | |
| 1 | 2 | |
| ЭСО.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. | |
| ЭСО.dproj | Файл проекта, связывает все файлы из которых состоит приложение. | |
| Unit3.pas | Файл программного модуля главной формы | |
| Unit2.pas | Файл программного модуля теоретической части проекта | |
| Unit4.pas | Файл программного модуля практической части проекта | |
| Unit5.pas | Файл программного модуля авторизация части проекта | |
| Unit6.pas | Файл программного модуля описание программы части проекта | |
| Unit7.pas | Файл программного модуля теста части проекта | |
| Unit3.dfm | Главная форма | |
| Unit2.dfm | Форма с теоретическим материалом | |
| Unit4.dfm | Форма с практическими заданиями | |
| Unit5.dfm | Форма с авторизацией | |
| Unit6.dfm | Форма с описанием программы | |
| Unit7.dfm | Форма с тестом | |
| Продолжение таблицы 3 | | | | |
| 1 | | | 2 | |
| тест.txt | | | Тест по повторенному материалу | |
| Setup.exe | | | Установочный пакет программы | |
| 1.docx…44.docx | | | Теоретический материал | |
| 1.docx…44.docx | | | Практический материал | |
| 1.jpg | | | Изображение | |
| 2.jpg | | | Изображение | |
| 3.jpg | | | Изображение | |
| 4.jpg | | | Изображение | |
| 5.jpg | | | Изображение | |
| 6.jpg | | | Изображение | |
| 7.jpg | | | Изображение | |
| 8.jpg | | | Изображение | |
| 9.jpg | | | Изображение | |
| 10.jpg | | | Изображение | |
| 11.jpg | | | Изображение | |
| 12.jpg | | | Изображение | |
| 13.jpg | | | Изображение | |

**4. Тестирование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения этапа написания программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчет о результатах тестирования представлен в таблице 4

Таблица 4- Отчет о результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор теста | Тест | Ожидаемый результат | Физический результат | Результат тестирование |
| 1 | Проверка входа  пользователя в приложение | Открытие главного окна | Открытие главного  окна | Выполнено |
| 2 | Проверка кнопки  «>>» | Открытие формы с  теоретическим материалом | Открытие формы с  теоретическим материалом | Выполнено |
| 3 | Проверка кнопки-  стрелки прокрутки  вниз/вверх | Переход вниз/вверх | Переход вниз/вверх | Выполнено |
| 4 | Проверка перехода на текстовый файл в формате .docx | Открытие текстового файла в формате .docx | Открытие текстового файла в формате .docx | Выполнено |
| 5 | Проверка главного меня на кнопку «Закрыть» | Закрытие приложения | Закрытие приложения | Выполнено |
| 6 | Проверка главного меня на кнопку «Теоретическая часть» | Открытие формы с теоретическим материалом | Открытие формы с теоретическим материалом | Выполнено |
| 7 | Проверка главного меня на кнопку «Практическая часть» | Открытие формы с практическими заданиями | Открытие формы с практическими заданиями | Выполнено |
| 8 | Проверка главного меня на кнопку «Тест» | Открытие формы с авторитизацией | Открытие формы с авторитизацией | Выполнено |
| 9 | Проверка главного меня на кнопку «Описание программы» | Открытие формы с описанием программы | Открытие формы с описанием программы | Выполнено |
| 10 | Проверка кнопки  «Далее» | Открытие формы с практическим материалом | Открытие формы с практическим материалом | Выполнено |
| Продолжение таблицы 4 | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 11 | Проверка кнопки  «Сгенерировать» | Генерация чисел | Генерация чисел | Выполнено |
| 12 | Проверка перехода на текстовый файл в формате .docx | Открытие текстового файла в формате .docx | Открытие текстового файла в формате .docx | Выполнено |
| 13 | Проверка кнопки  «Пройти тест» | Открытие формы с авторитизацией | Открытие формы с авторитизацией | Выполнено |
| 14 | Проверка кнопки  «Начать тест» | Открытие формы с тестом | Открытие формы с тестом | Выполнено |
| 15 | Проверка кнопки «Вариант ответа» | Выбор варианта ответа | Выбор варианта ответа | Выполнено |
| 16 | Проверка кнопки «Далее» | Переход на следующий вопрос | Переход на следующий вопрос | Выполнено |

При разработке программного продукта было решено множество проблем, например, не осуществлялся переход к следующему изображению по нажатию стрелки или открывался доступ к последнему квесту, когда был пройден только первый, поэтому в будущем пользователь не столкнется с данными проблемами.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Цель данного проекта заключается в обобщении знаний и подготовки к квалификационному экзамену.

Создаваемое приложение будет рассчитано на учащихся, проходивших профессиональную подготовку рабочих по профессии «Оператор ЭВМ ».

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на разных классах ЭВМ и работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

* процессор AMD Ryzen 7 4700U with Radeon Graphics, 2000 МГц, ядер: 8, логических процессоров: 8
* объем ОЗУ 32Гб;
* графический адаптер AMD Radeon Graphics Processor (0x1636), Advanced Micro Devices, Inc.-совместимый;
* операционная система Windows 10.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Установщик.exe. Появится окно установки приложения “ Электронное средство обучения для учащихся профессиональной подготовки рабочих по профессии «Оператор ЭВМ ”.

Затем достаточно следовать приведенной инструкции установки приложения.

**5.3 Выполнение программы**

**5.3.1 Запуск программы**

Данную программу можно запустить различными способами. Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием “ Электронное средство обучения для учащихся профессиональной подготовки рабочих по профессии «Оператор ЭВМ»

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\ Электронное средство обучения для учащихся профессиональной подготовки рабочих по профессии «Оператор ЭВМ).

По подготовленным тестам будет осуществляться функциональное и полное тестирование программного продукта. Отчет о результатах тестирования будет представлен в 4 разделе пояснительной записки.

**5.3.2 Инструкции по работе с программой**

После запуска приложения на экране нас встречает заставка на рисунок 2.

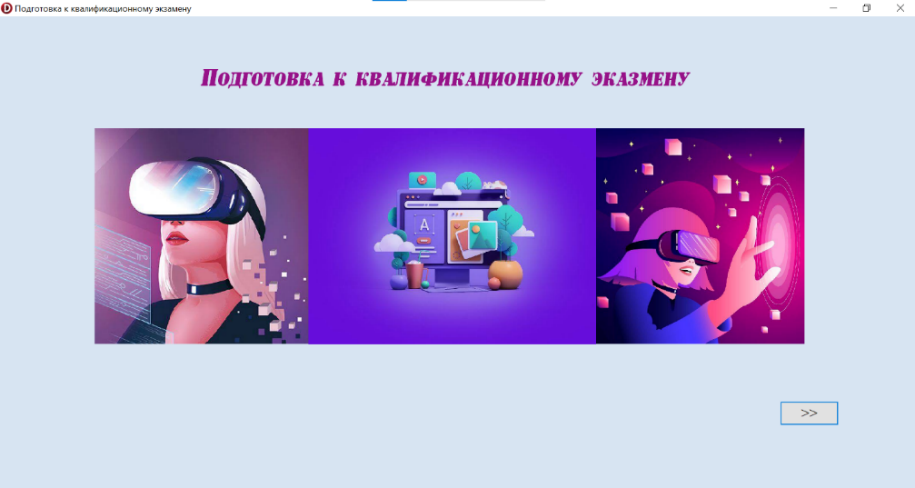


Рисунок 2 – Заставка

При нажатии на кнопку “>>” будет осуществлен переход на форму с теоретическим материалом, представлен на рисунке 3, где по нажатию на тему можно перейти на текстовый файл в формате .docx, представлен на рисунке 4.

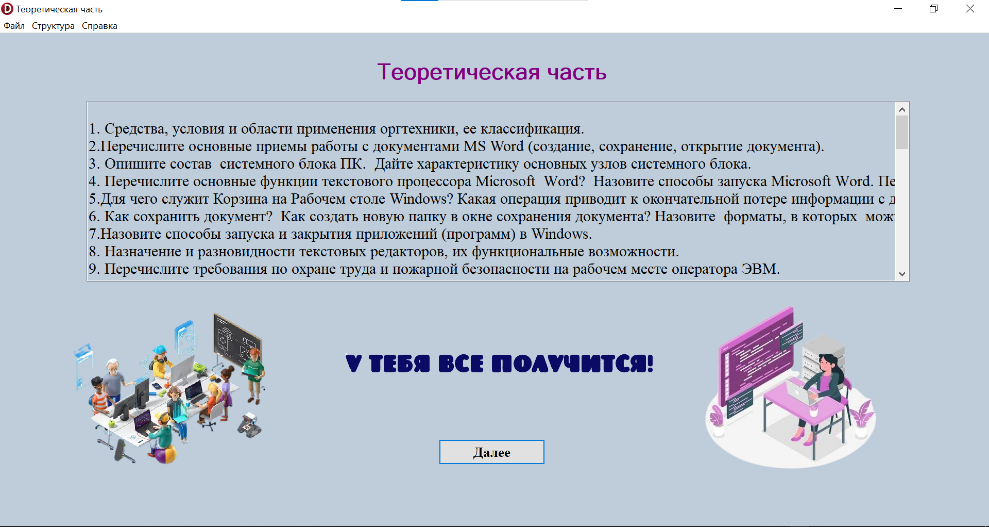
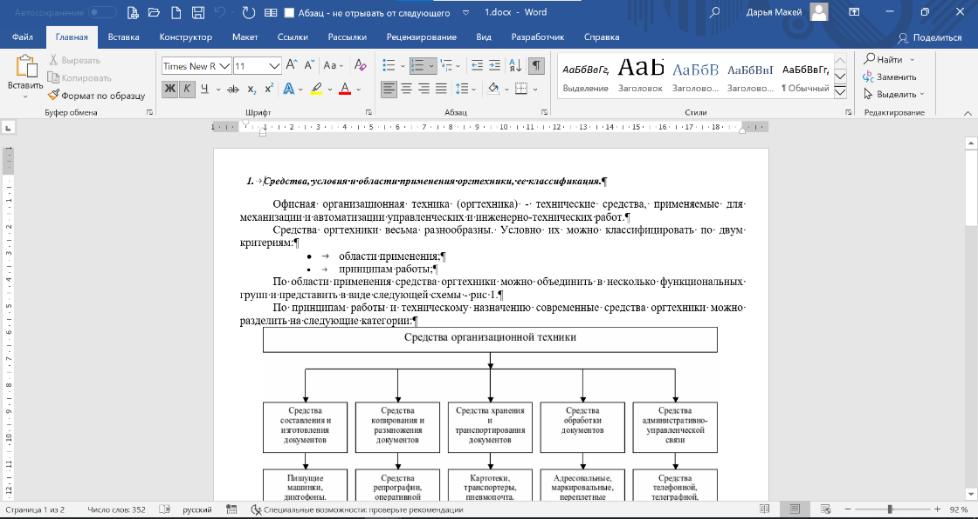


Рисунок 3-Теоретический материал

Рисунок 4- Текстовый файл в формате .docx

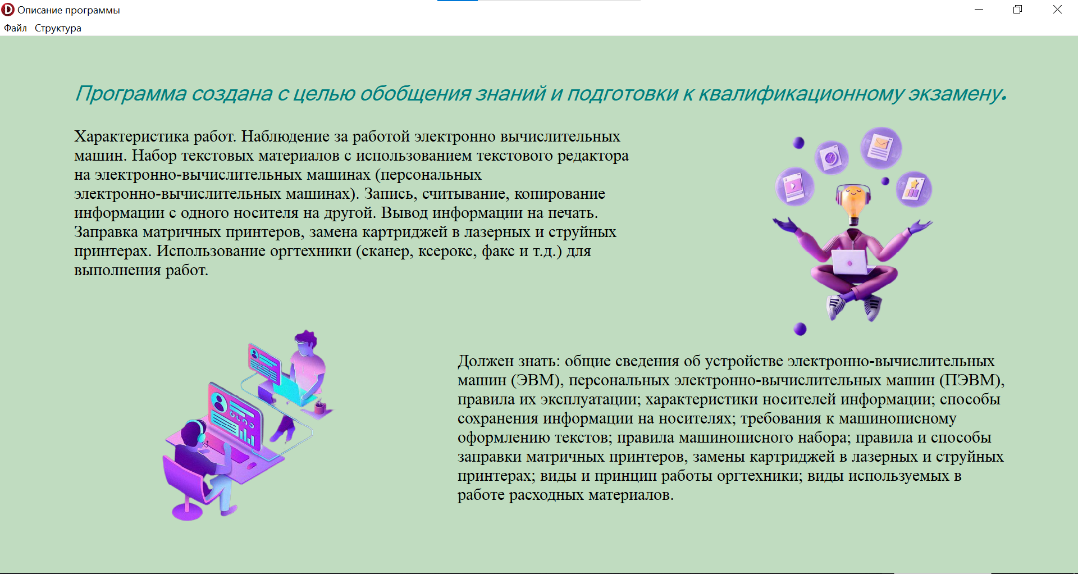
По нажатию на кнопку “Справка ” в главном меню можно выбрать описание программы, перейдя на новую форму рисунок 4.

Рисунок 5-Описание программы

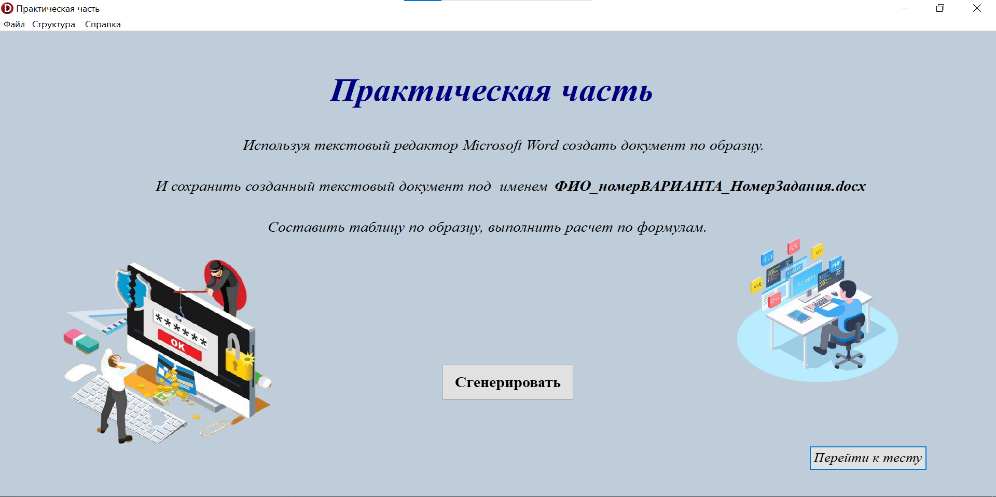
Вернуться назад можно через главное меню. Вернувшись назад на форму с теоретическим материалом, нажав на кнопку «Далее». Перед нами форма с практическими заданиями, представлена на рисунке

Рисунок 6-Практические задания

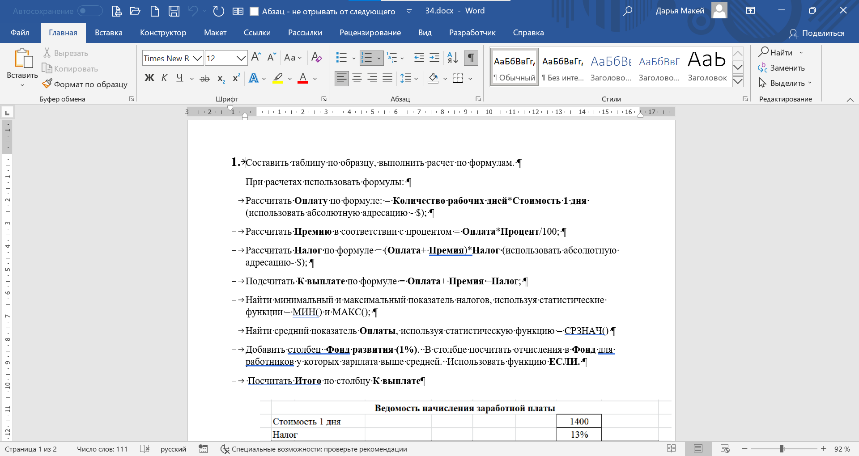
Нажав на кнопку «Сгенерировать», нам случайным выпадает число с задание, по которому можно перейти на текстовый файл в формате .docx на рисунке 7.

Рисунок 7-Практическое задание в текстовым файле в формате .docx

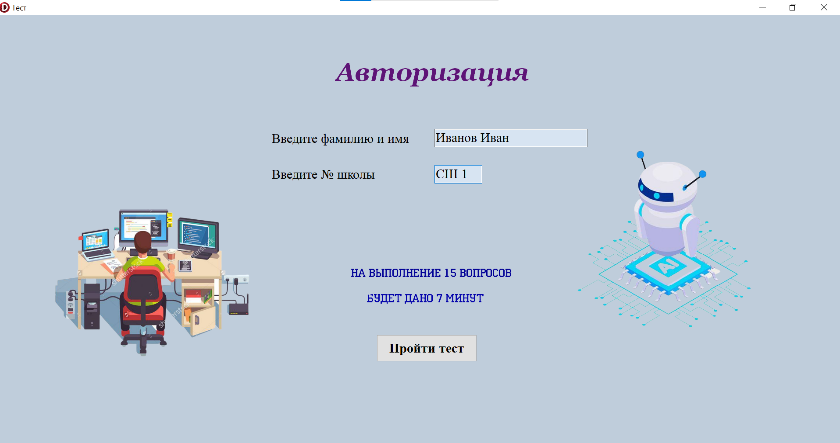
Нажав на кнопку «Пройти тест», переходим на форму с авторизацией, представлен на рисунке 8.

Рисунок 8-Авторизация.

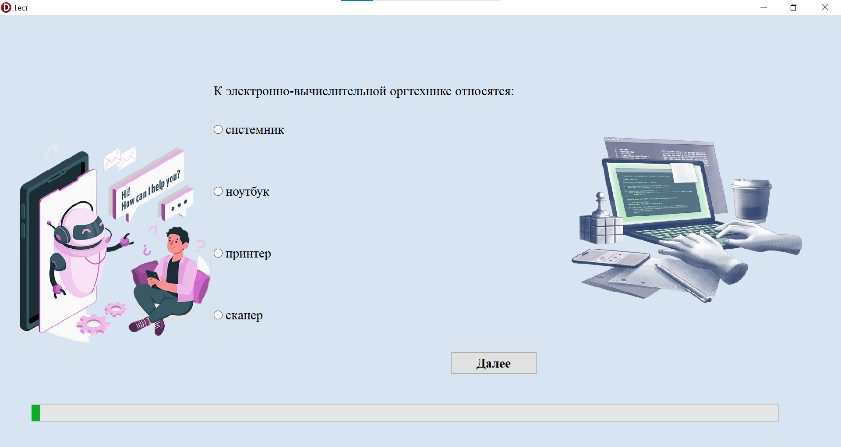
Начать тест можно по кнопке «Пройти в тест», после чего мы оказываемся на форме с тестом. Начав тест перед нами вопрос и ответы, можно посмотреть на рисунке 9.

Рисунок 9-Тест

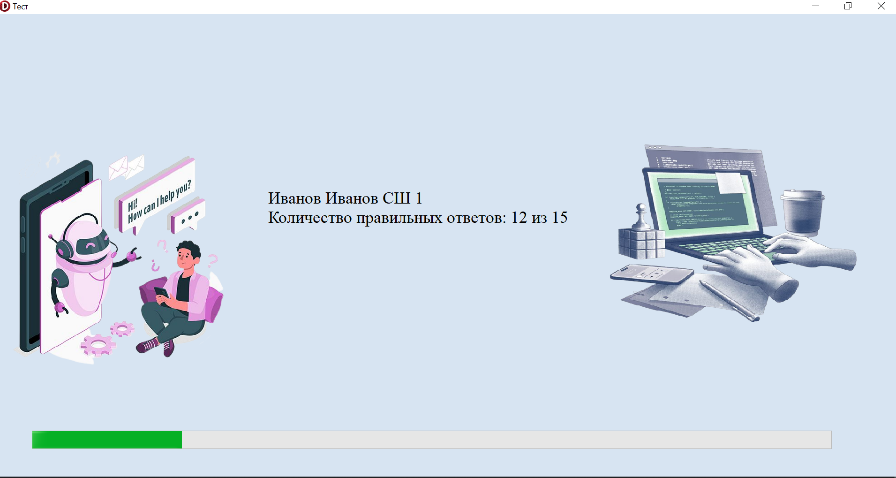
После завершение теста перед нами буду выведены результаты, представленные на рисунке 10

Рисунок 10-Результаты выполнения теста

**Заключение**

Целью данного проекта заключалось обобщение знаний и подготовка к квалификационному экзамену.

В поставленной задаче был реализован простой и понятный пользовательский интерфейс.

В ходе тестирования все исключительные ситуации были обработаны. Проект работает без сбоев и ошибок.

В процессе разработки программного продукта я научился создавать динамические компоненты на форме, проработал их взаимодействие, закрепил умение создания собственных процедур, научился обрабатывать все исключительные ситуации.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

**Приложение А**

Листинг программы

unit Unit3;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.jpeg,Shellapi,

Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

TForm3 = class(TForm)

Image1: TImage;

Image2: TImage;

Image3: TImage;

Button1: TButton;

Image4: TImage;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

uses Unit1, Unit2;

procedure TForm3.Button1Click(Sender: TObject);

begin

Form3.Hide;

Form2.Show;

end;

end.

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, Vcl.StdCtrls, ShellApi,

Vcl.Imaging.pngimage, Vcl.ExtCtrls;

type

TForm2 = class(TForm)

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

ListBox1: TListBox;

Label1: TLabel;

Button1: TButton;

Image1: TImage;

Image2: TImage;

Image3: TImage;

procedure N2Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure ListBox1DblClick(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

uses Unit1, Unit3, Unit4, Unit5, Unit6;

procedure TForm2.Button1Click(Sender: TObject);

begin

Form2.Hide;

Form4.Show;

end;

procedure TForm2.FormCreate(Sender: TObject);

var s, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13, s14, s15, s16, s17, s18, s19, s20, s21, s22, s23, s24:string ; s25, s26, s27, s28, s29, s30, s31, s32, s33, s34, s35, s36, s37, s38, s39, s40, s41, s42,s43, s44:string ;

begin

s1:='1. Средства, условия и области применения оргтехники, ее классификация.';

s2:='2. Перечислите основные приемы работы с документами MS Word (создание, сохранение, открытие документа). ';

s3:='3. Опишите состав системного блока ПК. Дайте характеристику основных узлов системного блока.';

s4:='4. Перечислите основные функции текстового процессора Microsoft Word? Назовите способы запуска Microsoft Word. Перечислите ленты главного меню. Какие группы кнопок расположены на Главной ленте? ';

s5:='5. Для чего служит Корзина на Рабочем столе Windows? Какая операция приводит к окончательной потере информации с диска?';

s6:='6. Как сохранить документ? Как создать новую папку в окне сохранения документа? Назовите форматы, в которых можно сохранить файлы, созданные в программе Word? Изложите способы открытия документа.';

s7:='7. Назовите способы запуска и закрытия приложений (программ) в Windows.';

s8:='8. Назначение и разновидности текстовых редакторов, их функциональные возможности.';

s9:='9. Перечислите требования по охране труда и пожарной безопасности на рабочем месте оператора ЭВМ.'

s10:='10. Какие основные программные компоненты включает Microsoft Office (описать все программы)?';

s11:='11. Объясните понятие междустрочного интервала. Опишите последовательность действий для установки межабзацных интервалов и последовательность действий для установки междустрочных интервалов.';

s12:='12. Для чего применяется программа PowerPoint? Перечислите способы создания презентаций.';

s13:='13. Опишите способы вставки таблицы в документ? Как изменить ширину столбца и высоту строки? Какие клавиши позволяют перемещать курсор по ячейкам таблицы? Как вставить строку или столбец в таблицу?';

s14:='14. С какими объектами работает программа Access? ';

s15:='15. Как вставить графический объект в документ? Как изменить цвет заливки фигуры? Как сделать заливку фигуры бесцветной? Опишите способы выделения фигуры. Как выбрать цвет и толщину линии?';

s16:='16. Как ввести исходные данные в Excel? Чем отличается ввод формульных данных Excel?';

s17:='17. Назовите способы вставки формулы в документ? Опишите назначение кнопок группы Структура на ленте Конструктор работы с формулами.';

s18:='18. Укажите области применения и круг задач, решаемых с помощью электронных таблиц. Основные понятия электронной таблицы Excel';

s19:='19. Перечислите основные блоки, входящие в состав компьютера. Опишите их основное назначение.';

s20:='20. Опишите основные операции редактирования электронной таблицы Excel: вставка и удаление строк, столбцов и листов рабочей книги.';

s21:='21. Опишите назначение следующих элементов текстового документа MS Word: оглавление, предметный указатель, сноска. Опишите методику их создания в текстовом редакторе MS Word.';

s22:='22. Опишите технологию добавления файлов в архив. Как просмотреть содержимое архивана примере WinRar?';

s23:='23. Опишите устройства внешней памяти, их разновидности, характеристики и принцип действия.';

s24:='24. Объясните понятие колонтитул. Перечислите способы вставки колонтитулов в документ. Как оформить разные колонтитулы на разных листах?';

s25:='25. Какие виды печатающих устройств вы знаете. Опишите их принципы работы.';

s26:='26. Что такое буфер обмена? Как скопировать или переместить объект с помощью буфера обмена?';

s27:='27. Как переместить и отредактировать картинку в Word';

s28:='28. Опишите назначение функций. Приведите примеры математических функций. Назовите функцию для вычисления модуля числа.';

s29:='29. Опишите функциональные особенности и назначение материнской платы.';

s30:='30. Расскажите о способах выделения, копирования и перемещения блоков ячеек электронной таблице';

s31:='31. Опишите функциональные особенности и назначение микропроцессора.';

s32:='32. Укажите области применения и круг задач, решаемых с помощью электронных таблиц.';

s33:='33. Перечислите и опишите функциональные особенности внешнец памяти ПК. Дайте им сравнительную характеристику.';

s34:='34. Как автоформатировать таблицу в Excel.';

s35:='35. Дать определение следующим понятиям: программа, программное обеспечение (ПО). Приведите классификацию программного обеспечения по области его использования.';

s36:='36. Перечислите все виды начертания шрифтов, используемых в Word. Как установить нужное начертание? На какой ленте находятся команды, необходимые для форматирования шрифта?';

s37:='37. Дать определение понятию оргтехника. Приведите классификацию оргтехники по области назначения.';

s38:='38. Перечислите способы установки параметров страницы в редакторе Word.';

s39:='39. Опишите назначение сканеров. Разновидности сканеров.';

s40:='40. Перечислите основные параметры форматирования абзаца. Перечислите все виды первой строки абзаца. Как их можно установить? Назовите способы выравнивания абзаца.';

s41:='41. Опишите технологию сканирования и распознавание документов.';

s42:='42. Расскажите о функциях в электронной таблице. Назовите способы вставки функции в формулу?';

s43:='43. Изложите порядок подключения принтера. Назовите правила настройки драйвера принтера, шрифтов и диспетчера печати.';

s44:='44. Назовите способы вставки диаграммы Excel. Как изменить тип построенной диаграммы в Excel.';

listbox1.Items.Append(s);

listbox1.Items.Append(s1);

listbox1.Items.Append(s2);

listbox1.Items.Append(s3);

listbox1.Items.Append(s4);

listbox1.Items.Append(s5);

listbox1.Items.Append(s6);

listbox1.Items.Append(s7);

listbox1.Items.Append(s8);

listbox1.Items.Append(s9);

listbox1.Items.Append(s10);

listbox1.Items.Append(s11);

listbox1.Items.Append(s12);

listbox1.Items.Append(s13);

listbox1.Items.Append(s14);

listbox1.Items.Append(s15);

listbox1.Items.Append(s16);

listbox1.Items.Append(s17);

listbox1.Items.Append(s18);

listbox1.Items.Append(s19);

listbox1.Items.Append(s20);

listbox1.Items.Append(s21);

listbox1.Items.Append(s22);

listbox1.Items.Append(s23);

listbox1.Items.Append(s24);

listbox1.Items.Append(s25);

listbox1.Items.Append(s26);

listbox1.Items.Append(s27);

listbox1.Items.Append(s28);

listbox1.Items.Append(s29);

listbox1.Items.Append(s30);

listbox1.Items.Append(s31);

listbox1.Items.Append(s32);

listbox1.Items.Append(s33);

listbox1.Items.Append(s34);

listbox1.Items.Append(s35);

listbox1.Items.Append(s36);

listbox1.Items.Append(s37);

listbox1.Items.Append(s38);

listbox1.Items.Append(s39);

listbox1.Items.Append(s40);

listbox1.Items.Append(s41);

listbox1.Items.Append(s42);

listbox1.Items.Append(s43);

listbox1.Items.Append(s44);

end;

procedure TForm2.ListBox1DblClick(Sender: TObject);

begin

ShellExecute(handle,'open',PWideChar('answers\'+inttostr(listbox1.ItemIndex)+'.docx'),'nil','nil',SW\_SHOWNORMAL)

end;

rocedure TForm2.N2Click(Sender: TObject);

begin

close;

Form2.Hide;

Меню.Show;

end;

procedure TForm2.N4Click(Sender: TObject);

begin

Form2.Hide;

Form4.Show;

end;

rocedure TForm2.N5Click(Sender: TObject);

begin

Form2.Hide;

Form5.Show;

end;

procedure TForm2.N7Click(Sender: TObject);

begin

Form2.Hide;

Form6.Show;

end;

end.

unit Unit4;

interface

uses 4

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, Vcl.StdCtrls, Vcl.CheckLst, ShellApi, Math,

Vcl.Imaging.jpeg, Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

TForm4 = class(TForm)

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

Label1: TLabel;

Button1: TButton;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Button2: TButton;

Label6: TLabel;

Image1: TImage;

Image2: TImage;

procedure N2Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Label5Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

uses Unit1, Unit2, Unit3, Unit5, Unit6;

procedure TForm4.Button1Click(Sender: TObject);

begin

Form4.Hide;

Form5.Show;

end;

procedure TForm4.Button2Click(Sender: TObject);

var

randomNumber: Integer;

begin

// Генерируем случайное число от 1 до 100

randomNumber := Random(44) + 1;

// Используйте полученное случайное число по вашему усмотрению

// Например, выведите его на метку (Label)

Label5.Caption := IntToStr(randomNumber);

end;

procedure TForm4.Label5Click(Sender: TObject);

begin

ShellExecute(handle,'open',PWideChar('practos\'+label5.Caption+'.docx'),'nil','nil',SW\_SHOWNORMAL)

end;

procedure TForm4.N2Click(Sender: TObject);

begin

close;

Form4.Hide;

Меню.Show;

end;

procedure TForm4.N4Click(Sender: TObject);

begin

Form4.Hide;

Form2.Show;

end;

procedure TForm4.N5Click(Sender: TObject);

begin

Form4.Hide;

Form5.Show;

end;

procedure TForm4.N7Click(Sender: TObject);

begin

Form4.Hide;

Form6.Show;

end;

end.

unit Unit5;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.pngimage, Vcl.ExtCtrls,

Vcl.StdCtrls;

type

TForm5 = class(TForm)

Label1: TLabel;

Image1: TImage;

Edit1: TEdit;

Edit2: TEdit;

Label2: TLabel;

Image2: TImage;

Button1: TButton;

Image3: TImage;

Image4: TImage;

Image5: TImage;

procedure Button1Click(Sender: TObject);

procedure Edit2Change(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

{$R \*.dfm}

uses Unit7;

procedure TForm5.Button1Click(Sender: TObject);

begin

Form5.Hide;

Form7.Show;

end;

procedure TForm5.Edit2Change(Sender: TObject);

begin

button1.Visible:=true;

end;

end.

unit Unit6;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.jpeg,

Vcl.ExtCtrls, Vcl.Menus, Vcl.Imaging.pngimage;

type

TForm6 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

Image1: TImage;

Image2: TImage;

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N6Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

implementation

{$R \*.dfm}

uses Unit2, Unit3, Unit4, Unit5;

procedure TForm6.N3Click(Sender: TObject);

begin

close;

end;

procedure TForm6.N4Click(Sender: TObject);

begin

Form6.Hide;

Form2.Show;

end;

procedure TForm6.N5Click(Sender: TObject);

begin

Form6.Hide;

Form4.Show;

end;

procedure TForm6.N6Click(Sender: TObject);

begin

Form6.Hide;

Form5.Show ;

end;

end.

unit Unit7;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls, Vcl.ComCtrls;

type

TForm7 = class(TForm)

Button1: TButton;

Image1: TImage;

Image2: TImage;

RadioGroup1: TRadioGroup;

Button2: TButton;

ProgressBar1: TProgressBar;

Timer1: TTimer;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form7: TForm7;

var

f: text;

s: string;

Nvern, ball, i, x: integer;

implementation

{$R \*.dfm}

uses Unit8, Unit5;

procedure TForm7.Button1Click(Sender: TObject);

begin

//Если выбран вариант ответа и не достигнут конец файла

if (RadioGroup1.ItemIndex>-1) and (not Eof(f)) then begin

if RadioGroup1.ItemIndex = Nvern-1 then ball:=ball+1; //Если выбранный вариант соответствует

RadioGroup1.Items.Clear; //номеру верного ответа то балл прибавляется

Repeat //и очищается поле для следующего вопроса

if (s[1]='-') then begin

delete(s,1,1);

RadioGroup1.Caption:=s;

end

else if s[1]='\*' then begin

delete(s,1,1);

Nvern:=StrToInt(s);

end

else RadioGroup1.Items.Add(s);

readln(f,s);

until (s[1]='-') or (Eof(f));

end

//Если конец файла достигнут, значит вопросы закончились

Else if Eof(f) then begin

delete(s,1,1);

Nvern:=StrToInt(s);

if RadioGroup1.ItemIndex = Nvern-1 then ball:=ball+1;

Label1.Caption:=(form5.Edit1.Text+' '+form5.Edit2.Text+#13+'Количество правильных ответов: '+IntToStr(ball)+' из 15'); //Вывод количества баллов

CloseFile(f);

RadioGroup1.Visible:=false;

Button1.Visible:=False;

Button2.Visible:=False;

Button1.Enabled:=False; //кнопка становится недоступной в завершении проекта

end;

end;

procedure TForm7.Button2Click(Sender: TObject);

begin

timer1.Enabled :=true

progressbar1.Visible:=True;

RadioGroup1.Visible:=True;

button2.Visible:=false;

button1.Visible:=true;

AssignFile( f ,'тест.txt');//связывается с разными файлами

reset(f); //Открываем файл для чтения

readln(f,s); //Считываем первую строку из файла

ball:=0; //изначально количество баллов 0

repeat

if (s[1]='-') then begin //Если первый символ строки ‘-‘ значит это вопрос

delete(s,1,1);

RadioGroup1.Caption:=s;

end

else if s[1]='\*' then begin //Если перв символ ‘\*’ значит это номер верного ответа

delete(s,1,1);

Nvern:=StrToInt(s);

end

else RadioGroup1.Items.Add(s); //Иначе это вариант ответа

readln(f,s); //Считываем следующую строку из файла

until (s[1]='-') or (Eof(f)); //Считывание и отправление вариантов ответов в RadiGroup до тех пор// пока не достигнут следующий вопрос или конец файла

end;

procedure TForm7.Timer1Timer(Sender: TObject);

begin

progressbar1.Position :=progressbar1.Position +1 ;

if progressbar1.position = 420 then

begin

timer1.Enabled :=false;

Showmessage('Время вышло');

close;

end;

end;

end.